

Implementasi Metode *Forward Chaining* untuk Menentukan Kenaikan Level pada *Game Finding Selais*

Reny Marissa

Jurusan Teknik Informatika, STMIK Amik Riau
reny_marissa@stmik-amik-riau.ac.id

Erlin

Jurusan Teknik Informatika, STMIK Amik Riau
erlin@stmik-amik-riau.ac.id

Abstrak

Game adalah media hiburan yang kerap menjadi pilihan bagi orang-orang yang ingin menghilangkan kejenuhan atas rutinitas sehari-hari. Perkembangan zaman sekarang ini mengakibatkan waktu yang bisa dimanfaatkan untuk melakukan permainan terbatas, sehingga bermain *game* diperangkat elektronik menjadi pilihan utama bagi mereka yang memiliki keterbatasan waktu, tempat dan lawan bermain *game*. Namun disisi lain masih banyak *game* yang mengandung kekerasan dan kurang mendidik serta sedikitnya *game* yang mengangkat karakteristik wilayah setempat. Oleh sebab itu penelitian ini bertujuan untuk membuat sebuah *game* yang mengangkat karakteristik daerah Riau dengan ikan selais sebagai maskot dari *game* yang dinamai *Finding Selais*. *Game finding selais* yang bertipe *adventure games* dengan *level up system* ini merupakan implementasi konsep *Artificial Intelligence*, khususnya metode *forward chaining* yang dimanfaatkan untuk menentukan kenaikan level. *Game* yang diaplikasikan pada platform desktop (Windows) ini hanya dapat dimainkan oleh seorang player pada sekali permainan. Untuk menghindari rintangan dan memakan hewan laut lainnya, player dapat melakukannya dengan menekan tombol Up dan Down pada keyboard. Level Up pada sistem ini menitik beratkan pada ikan selais yang harus menyelesaikan perjalanannya hingga menemui kumpulan ikan selais lainnya. *Game finding selais* juga menggunakan *game desain dokumen* sebagai cetak biru yang kemudian menjadi panduan bagi penulis dalam proses pengembangan *game* tersebut.

Kata Kunci : *forward chaining*, desain dokumen, *game*, *finding selais*

1. Pendahuluan

Game adalah media hiburan yang kerap menjadi pilihan bagi orang-orang yang ingin menghilangkan kejenuhan atas rutinitas sehari-hari. Ada banyak macam permainan yang tersedia. Permainan

konvensional biasa dimainkan oleh orang-orang yang memiliki waktu luang untuk melakukannya. Akan tetapi melihat pada perkembangan zaman sekarang ini mengakibatkan waktu yang bisa dimanfaatkan untuk melakukan permainan tersebut terbatas, sehingga bermain *game* diperangkat elektronik menjadi pilihan utama bagi mereka yang memiliki keterbatasan waktu, tempat dan lawan bermain *game*.

Game komputer berkembang dengan jenis yang beragam, mulai dari *game* yang hanya dapat dimainkan oleh satu *user* saja hingga *game* yang dapat dimainkan oleh beberapa *user* sekaligus. Perkembangan industri *game* banyak sekali melibatkan peran ilmu di bidang matematika dan informatika.

Beberapa peneliti terdahulu telah melakukan penelitian tentang pembuatan *game* dengan menggunakan berbagai macam teknik dan metoda. Diantara peneliti tersebut adalah Putra et al., [1] yang membuat *Game Chicken Roll* dengan menggunakan metode *forward chaining* untuk proses review dan perpindahan level. Zainuddin [2] merancang dan membuat program permainan Quad dengan metode *forward chaining*. Selanjutnya Estrada [3] merancang dan membuat *game* Minimax pada Tic Tac Toe dengan mengimplementasikan Pohon N-Ary dalam artificial intelligence. Andiansyah [4] juga telah berhasil membuat *game* Otak Atik dengan menggunakan perancangan *game* desain dokumen dan aset grafis. Namun sejauh pengetahuan penulis belum ada *game* yang mengangkat karakteristik wilayah setempat, khususnya Riau. Hal inilah yang melatarbelakangi penulis melakukan suatu penelitian pembuatan *game finding selais* yang merupakan ikon wilayah Riau dengan menggunakan metode *forward chaining*.

2. Metode *Forward Chaining*

Forward chaining merupakan salah satu metode inferensi yang dipakai untuk mendapatkan konklusi menggunakan penalaran maju. Cara untuk menggambarkan *forward chaining* adalah dengan memulai penalaran dari fakta-fakta (data) yang mencari *rules* atau aturan yang cocok untuk mendapatkan konklusi dari fakta tersebut [5]. Contoh

sederhana dari metode *forward chaining* seperti berikut ini: misalkan anda sedang mengemudi dan tiba-tiba anda melihat mobil polisi dengan cahaya kelap-kelip dan bunyi sirine. Dengan *forward chaining* mungkin anda akan berkesimpulan bahwa polisi ingin anda atau seseorang untuk berhenti. Itu adalah fakta awal yang mendukung dua kemungkinan konklusi. Jika mobil polisi membuntuti dibelakang anda atau polisi melambatkan tangan memberhentikan anda, maka kesimpulan lebih lanjut adalah polisi ingin anda yang berhenti.

Kelebihan metode *forward chaining* ini adalah data baru dapat dimasukkan ke dalam table *database* inferensi dan kemungkinan untuk melakukan inferensi *rules*, sehingga dalam implementasinya *forward chaining* sangat membantu *developer* aplikasi dalam membangun sebuah sistem. Jika *developer* ingin menambah beberapa kondisi dan aturan, *developer* tidak perlu membongkar lagi kode program dari awal. Tabel 1 dibawah adalah karakteristik *forward chaining*.

Tabel 1. Karakter *forward chaining*

<i>Forward Chaining</i>
Perencanaan, <i>Monitoring</i> , <i>Control</i> . Disajikan untuk masa depan. Data memandu, penalaran dari bawah ke atas. Bekerja kedepan untuk mendapatkan solusi apa yang mengikuti fakta. <i>Breadth first search</i> dimudahkan.

Metode *forward chaining* disebut juga *data driven* karena mesin inferensi menggunakan informasi yang ditentukan oleh *user* untuk memindahkan keseluruhan jaringan dari logika 'AND' dan 'OR' sampai sebuah terminal ditentukan sebagai obyek. Bila mesin inferensi tidak dapat menentukan obyek maka mesin inferensi akan meminta informasi lain.

3. Game Desain Dokumen

Game desain dokumen adalah kumpulan dokumen-dokumen yang digunakan *game designer* untuk menginformasikan mengenai *game* yang didesain,

proses ini mengubah ide yang tadinya abstrak menjadi rencana tertulis [6].

Game desain dokumen merupakan sebuah cetak biru dari sebuah *game* yang kemudian menjadi panduan bagi seluruh tim *developer* dalam proses pengembangan *game*. Sebuah dokumen *game* bersifat dinamis dengan kata lain bisa terus di-*update*. Namun demikian, beberapa bagian dari *game* desain dokumen harus bisa tersusun secara konkret pada tahapan awal, hal ini menjadi kerangka utama dari keseluruhan konsep *game* yang tidak bisa lagi diubah. Kerangka utama umumnya berupa *gameplay* atau mekanisme utama dari sebuah *game*. Selain kerangka *gameplay* dan mekanisme, masih banyak tugas-tugas yang terkait dengan *game* desain seperti: *level design* dan *user interface design*. Dengan adanya *game* desain dokumen yang detail akan mampu memberikan gambaran lengkap bagi seluruh tim sehingga proses pengembangan *game* lebih efektif dan efisien.

Game desain menjadi sangat penting karena proses ini yang kemudian menentukan bentuk, keunikan, serta fungsi dari sebuah *game*. Setiap *game* adalah sebuah karya dan selayaknya memiliki bentuk, keunikan, dan fungsi tertentu, bukan sekedar mengikuti apa yang sudah ada sebelumnya [7].

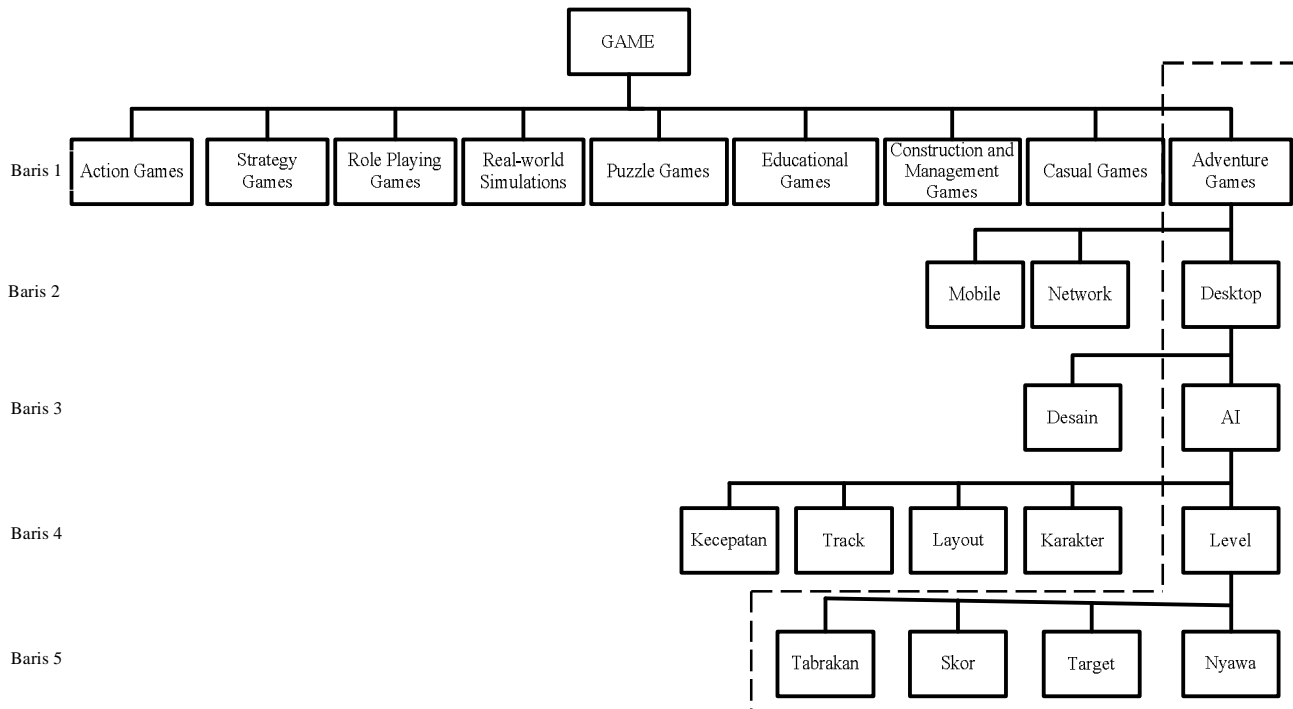
Kemudian proses berikutnya dalam pengembangan *game*, salah satunya adalah membuat aset grafis. Sebuah *game* tidak lengkap rasanya jika *game* tersebut tidak memiliki tampilan grafis yang menarik.

Berikut ini komponen-komponen utama dalam penyusunan *Game Desain Dokumen*.

1. *Game Overview*.
2. *Level Design*.
3. *World Design*.
4. *User Interface Design*.
5. *Content Design*.
6. *System Design*.

4. Kerangka Konsep Penelitian

Kerangka konsep penelitian dibuat dengan tujuan mendeskripsikan penelitian yang dikerjakan. Gambar 1 dibawah ini adalah kerangka konsep penelitian *game finding selais*.



Gambar 1. Kerangka konsep penelitian

Keterangan gambar kerangka konsep penelitian diatas ialah sebagai berikut:

- Garis putus-putus: penelitian yang dikerjakan.
- Baris 1: Game mempunyai Sembilan jenis (*Action Games, Strategy Games, Role Playing Games, Real-World Simulations, Puzzle Games, Educational Games, Construction and Managemen Games, Casual Games, Adventure Games*).
- Baris 2 : Penelitian ini fokus pada pengembangan *game* untuk *desktop*.
- Baris 3 : Penelitian ini lebih mengutamakan pengembangan pada bidang AI (*Artificial Intelligence*) khususnya dengan menggunakan *forward chaining*.
- Baris 4 : Konsep AI dimanfaatkan untuk proses *leveling*. Memanfaatkan *forward chaining* untuk menentukan seorang *player* dapat melanjutkan ke *level* berikutnya atau tidak.
- Baris 5 : Variabel yang dijadikan acuan sebagai bahan *input* untuk konsep AI dengan *forward chaining*.

Berikut ini adalah variabel yang digunakan sebagai parameter dalam *forward chaining* untuk pengembangan *game finding selais*:

Tabel 2. Variabel aturan *game finding selais*

No	Kode Rule	Kondisi	Keterangan
1.	A01	[0, 1, 2, 3]	Nyawa
2.	AXX

Tabel 3. Variabel hasil *game finding selais*

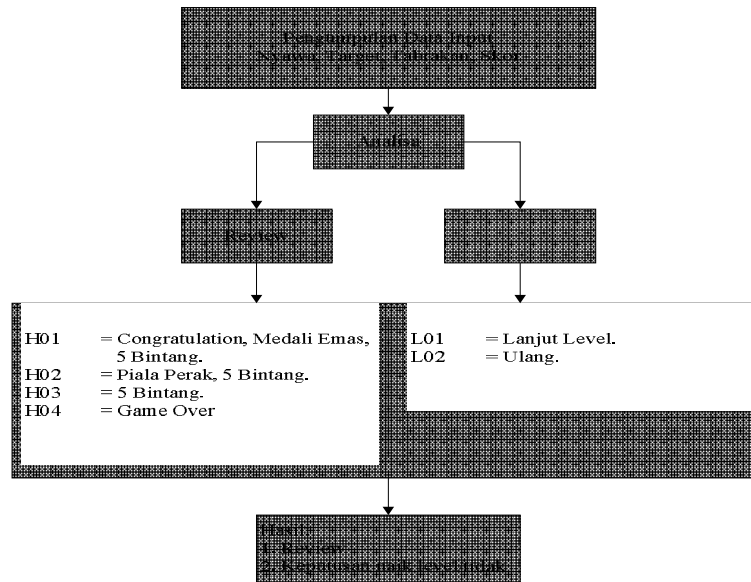
No	Kode Rule	Kondisi
1.	H01	<i>Congratulation</i> , Medali Emas, 5 Bintang.
2.	HXX	...

Tabel 4. Variabel level *game finding selais*

No	Kode Rule	Kondisi
1.	L01	Lanjut <i>Level</i>
2.	LXX	...

4.1. Proses Representasi Cara Kerja Game

Proses representasi pengetahuan dilakukan dengan cara mengumpulkan kondisi untuk memperoleh hasil. Berdasarkan data *input* yang diperoleh maka dilakukan analisis untuk menentukan *review* apa yang didapat dalam setiap levelnya serta layak atau tidaknya seorang *player* untuk naik ke *level* berikutnya dalam *game*. Gambar 2 di bawah ini merupakan proses representasi cara kerja *game*.



Gambar 2. Proses representasi cara kerja *game* (modifikasi dari putra, 2013)

Hasil representasi pengetahuan kondisi aturan disajikan pada Tabel 5, representasi pengetahuan *review* disajikan pada Tabel 6, sedangkan representasi pengetahuan kenaikan level disajikan pada Tabel 7.

Tabel 5. Representasi kondisi

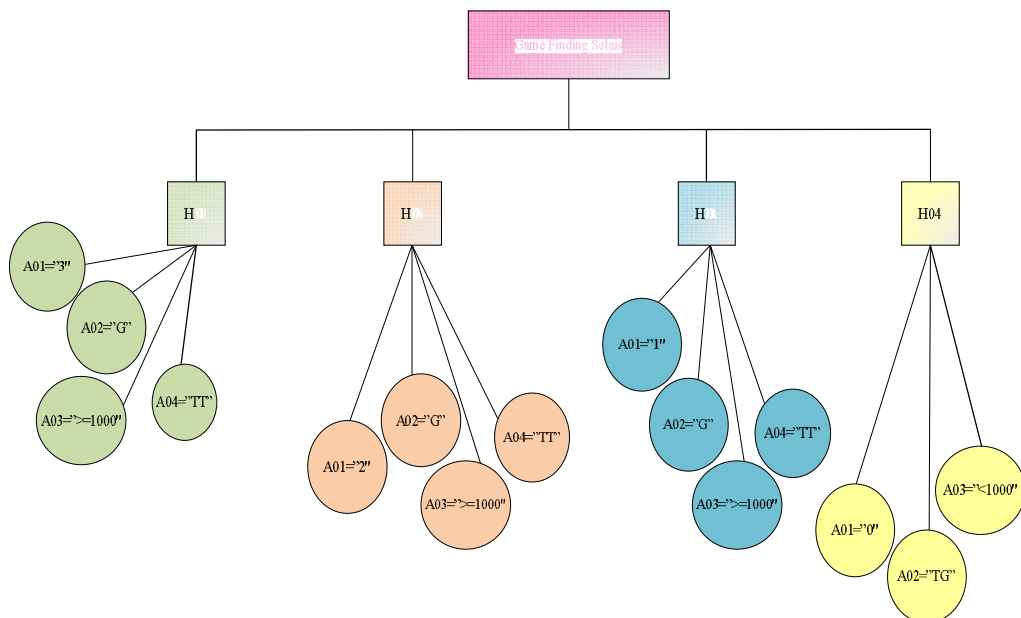
No	Kondisi	Kode Rule	Keterangan
1.	[0, 1, 2, 3]	A01	Nyawa
2.	[G, TG]	A02	Target
3.	[=>1000, <1000]	A03	Skor
4.	[T, TT]	A04	Tabrakan

Tabel 6. Representasi *review*

No	Kondisi	Kode
1.	Congratulations, Medali Emas, 5 Bintang.	H01
2.	Medali Perak, 5 Bintang.	H02
3.	5 Bintang.	H03
4.	Game Over.	H04

Tabel 7. Representasi level

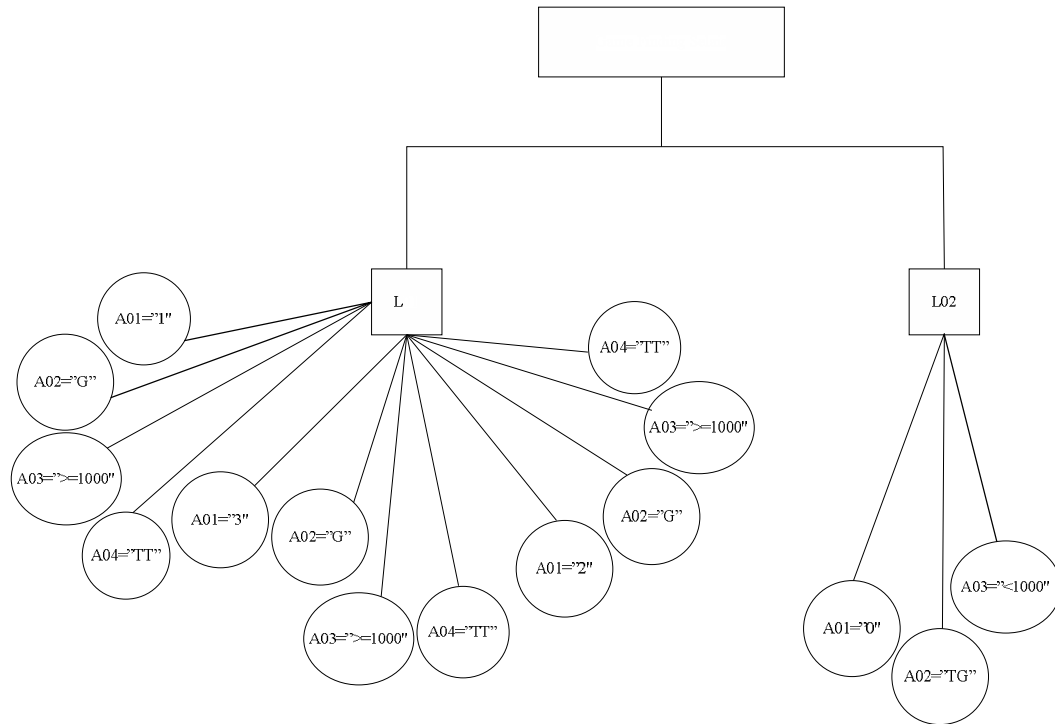
No	Kode Level	Keterangan
1.	L01	Lanjut Level
2.	L02	Ulang



Gambar 3. Inference Tree Review dan Kondisi

Review dan Hasil direpresentasikan menjadi sebuah *inference tree* yang sangat membantu dalam pembentukan aturan, selanjutnya akan diproses oleh mesin inferensi (*inference engine*). Gambar 3 di atas adalah *Inference Tree* dari Review dan Kondisi.

Pada Gambar 4 di bawah ini menunjukkan *inference tree* dari hasil dan kondisi yang menentukan kondisi seorang *player* dapat naik level atau tidak.



Gambar 4. *Inference Tree* Hasil dan Kondisi

6. Hasil dan Pembahasan

Hasil implementasi aplikasi *game finding selais* dapat dijelaskan melalui gambar-gambar hasil *print screen* dari *game* tersebut. Berikut ini tampilan-tampilan aplikasi *game finding selais*.

1. Tampilan Menu Utama



Gambar 5. Tampilan menu utama

Pada saat pengguna baru menjalankan aplikasi *game* maka pengguna akan masuk ke tampilan menu utama seperti yang terlihat pada gambar 5 diatas.

Pada gambar di atas terdapat tombol-tombol yang bisa digunakan untuk mengarahkan pengguna ke

tampilan lainnya. Pengguna dapat mengklik tombol *play game* untuk memulai permainan, tombol *options* untuk menghidupkan atau mematikan suara pada *game*, tombol *credit* untuk mengetahui kredit pengembang *game*, dan tombol *exit* untuk keluar dari aplikasi.

2. Tampilan Options

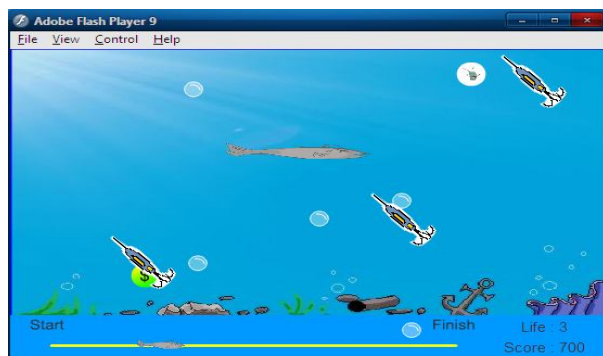
Tampilan *options* seperti gambar 6 dibawah menampilkan tombol *ON* untuk menghidupkan *sound* atau suara pada *game*. Tombol *OFF* untuk mematikan *sound* pada *game*. Dan tombol *main menu* maka akan kembali ke tampilan menu utama.



Gambar 6. Tampilan *options*

3. Tampilan *Play Game*

Gambar 7 dibawah adalah tampilan saat permainan sedang berlangsung, terdapat informasi *score* dan *life* pemain, serta tombol *quit* untuk keluar dari permainan.



Gambar 7. Tampilan permainan berlangsung

4. Tampilan Hasil Permainan

Salah satu tampilan hasil permainan diperlihatkan pada gambar 8 dibawah. Hasil *review* akan ditampilkan sesuai dengan kondisi yang dipenuhi oleh pemain. Adapun *reward* atau penghargaan yang tersedia adalah ucapan *congratulation*, medali emas, medali perak dan lima bintang. Pemain bisa naik ke level berikutnya.



Gambar 8. Tampilan pemain memperoleh *review* H01

5. Tampilan *Game Over*



Gambar 9. Tampilan *game over*

Gambar 9 diatas adalah tampilan *game over* yang akan muncul jika pemain gagal membawa ikan selais sampai. Pengguna dapat menekan tombol *try again* untuk bermain lagi.

7. Pengujian Sistem

Pengujian merupakan hal terpenting yang bertujuan untuk menentukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang akan diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan.

Pengujian yang dilakukan yaitu pengujian *black box* dan pengujian beta. Pengujian *black box* digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Pengujian beta digunakan untuk mengetahui tanggapan *user* terhadap aplikasi *game*, dengan melakukan kuisioner.

7.1. Pengujian *Black Box*

Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian *alpha*. Metode yang digunakan dalam pengujian ini adalah pengujian *black box* yang berfokus pada persyaratan fungsional dari sistem yang dibangun. Kasus dan hasil pengujian adalah sebagai berikut:

a. Pengujian Menu Utama

Tabel 8. Pengujian menu utama

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Menu Utama	Memilih Tombol <i>Play Game</i>	Menampilkan Area Permainan	[√] Berhasil [] Tidak Berhasil
	Memilih Tombol <i>Options</i>	Menampilkan pilihan <i>on/off sound</i> permainan	
	Memilih Tombol <i>Credit</i>	Menampilkan kredit pengembang	
	Memilih Tombol <i>Keluar</i>	Keluar dari permainan	

b. Pengujian *Play Game*

Tabel 9. Pengujian *play game*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
	Ketika pemain memilih tombol <i>Play Game</i>	Pemain berada di area permainan	[√] Berhasil [] Tidak Berhasil
	Informasi perolehan skor	Skor yang diperoleh oleh pemain ditampilkan	

Menu <i>Play Game</i>	Informasi nyawa yang tersisa	Menampilkan nyawa yang tersisa	
	Ketika ikan selais menyentuh <i>movie clip</i> plankton	Skor pemain bertambah 100	
	Ketika ikan selais menyentuh <i>movie clip</i> koin	Skor pemain bertambah 200	
	Ketika ikan selais menyentuh <i>movie clip</i> mata pancing	Nyawa pemain berkurang 1	
	Ketika ikan selais tiba digaris <i>finish</i>	Permainan selesai	

c. Pengujian *Review* Hasil PermainanTabel 10. Pengujian *review* hasil permainan

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
<i>Re-view</i>	Jika nyawa tersisa 3 dan skor ≥ 1000	Menampilkan <i>congratulation</i> , medali emas, lima bintang, skor, nyawa dan tombol <i>next level</i> .	[√] Berhasil [] Tidak Berhasil
	Jika nyawa tersisa 2 dan skor ≥ 1000	Menampilkan medali perak, lima bintang, skor, nyawa dan tombol <i>next level</i> .	
	Jika nyawa tersisa 1 dan skor ≥ 1000	Menampilkan lima bintang, skor, nyawa dan tombol <i>next level</i> .	
	Jika nyawa = 0	Menampilkan <i>Game Over</i> dan tombol <i>try again</i>	

d. Pengujian *Options*Tabel 11. Pengujian *options*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
<i>Options</i>	Ketika pemain memilih Tombol <i>Options</i>	Menampilkan pilihan <i>on/off</i> <i>sound</i> permainan	[√] Berhasil [] Tidak Berhasil

	Tombol <i>main menu</i>	Kembali ke menu utama	
--	-------------------------	-----------------------	--

e. Pengujian *Credit*Tabel 12. Pengujian *credit*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
<i>Credit</i>	Ketika pemain memilih Tombol <i>Credit</i>	Menampilkan kredit pengembang	[√] Berhasil [] Tidak Berhasil
	Tombol <i>main menu</i>	Kembali ke menu utama	

f. Pengujian *Credit*Tabel 13. Pengujian menu *exit*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
<i>Menu Exit</i>	Ketika pemain memilih Tombol <i>Exit</i>	Keluar dari aplikasi	[√] Berhasil [] Tidak Berhasil

7.2. Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana aplikasi diuji secara langsung ke lapangan dengan menggunakan kuisioner mengenai tanggapan *user* terhadap *game* yang telah dibangun. Kuisioner disebarakan menggunakan teknik *Simple Random Sampling* dimana anggota *sample* dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi. Kuisioner disebarakan kepada 30 orang siswa-siswi sekolah menengah atas sederajat rentang usia antara 15 tahun sampai 17 tahun. Kuisioner terdiri dari lima pertanyaan dengan sistem penskoran menggunakan skala pengukuran teknik *Likert*, berikut adalah skor untuk jawaban kuisioner:

Tabel 14. Skala kuisioner

Skala Jawaban	Keterangan	Skor
S	Setuju	3
BS	Biasa saja	2
TS	Tidak setuju	1

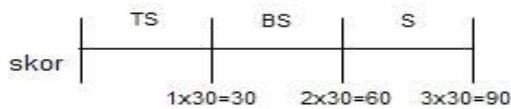
Data yang diperoleh dari pemberian kuisioner kepada responden dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan scoring setiap jawaban dari responden, analisis yang dapat dilakukan yaitu sebagai berikut:

Jumlah skor ideal untuk seluruh item = 3 x jumlah responden

Rata-rata skor dalam persentase:

$(\text{jumlah skor} : \text{jumlah skor ideal}) \times 100\%$
jumlah skor ideal = $3 \times 30 = 90$

Secara kontinum dapat digambarkan seperti gambar 10.



Gambar 10. Skor Kontinum

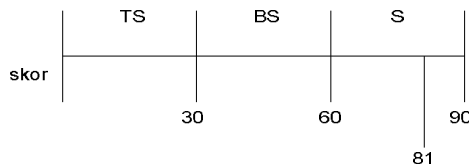
Adapun pertanyaan dan hasil observasi yang telah dibagikan kepada responden adalah sebagai berikut:

1. Apakah game ini dapat memberikan informasi kepada anda bahwa ikan selais merupakan salah satu ikon Provinsi Riau?

Tabel 15. Hasil survey pertanyaan 1

Kode Jawaban	S	SB	TS
Frequensi Jawaban	22	7	1
Skor	$22 \times 3 = 66$	$7 \times 2 = 14$	$1 \times 1 = 1$
Jumlah Skor	81		
persentase	$81/90 * 100\% = 90\%$		

Garis Kontinum



Gambar 11. Garis kontinum kuisioner no. 1

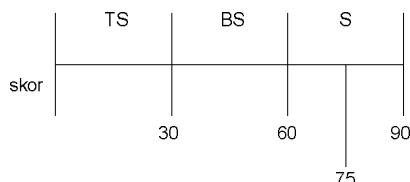
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap bisa tidaknya aplikasi ini memberikan informasi tentang ikan selais sebagai ikon Provinsi Riau adalah 81 dari yang diharapkan adalah 90, atau dapat dikategorikan setuju.

2. Menurut anda apakah aplikasi game finding selais ini mudah digunakan?

Tabel 16. Hasil survey pertanyaan 2

Kode Jawaban	S	SB	TS
Frequensi Jawaban	17	11	2
Skor	$17 \times 3 = 51$	$11 \times 2 = 22$	$2 \times 1 = 2$
Jumlah Skor	75		
persentase	$75/90 * 100\% = 83,3\%$		

Garis Kontinum



Gambar 12. Garis kontinum kuisioner no. 2

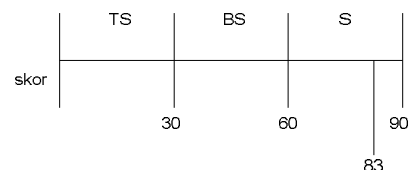
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap mudah atau tidaknya pengguna mengoperasikan aplikasi adalah 75 dari yang diharapkan adalah 90, atau dapat dikategorikan setuju.

3. Menurut anda apakah permainan dalam game finding selais ini menarik untuk dimainkan?

Tabel 17. Hasil survey pertanyaan 3

Kode Jawaban	S	SB	TS
Frequensi Jawaban	23	7	0
Skor	$23 \times 3 = 69$	$7 \times 2 = 14$	$0 \times 1 = 0$
Jumlah Skor	83		
persentase	$83/90 * 100\% = 92,2\%$		

Garis Kontinum



Gambar 13. Garis kontinum kuisioner no. 3

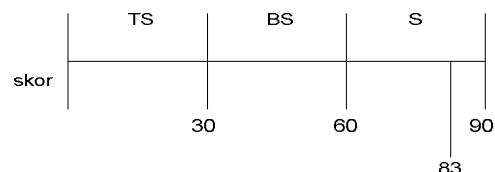
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap bisa tidaknya aplikasi ini menjadikan pengguna tertarik dengan permainan finding selais adalah 83 dari yang diharapkan adalah 90, atau dapat dikategorikan setuju.

4. Apakah anda tertarik untuk memainkan lagi game finding selais ini?

Tabel 18. Data hasil survey pertanyaan 4

Kode Jawaban	S	SB	TS
Frequensi Jawaban	23	7	0
Skor	$23 \times 3 = 69$	$7 \times 2 = 14$	$0 \times 1 = 0$
Jumlah Skor	83		
persentase	$83/90 * 100\% = 92,2\%$		

Garis Kontinum



Gambar 14. Garis kontinum kuisioner no. 4

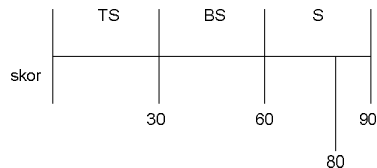
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap menarik atau tidaknya aplikasi ini adalah 83 dari yang diharapkan adalah 90, atau dapat dikategorikan setuju.

5. Apakah tampilan game ini menarik?

Tabel 19. Data hasil survey pertanyaan 5

Kode Jawaban	S	SB	TS
Frekuensi Jawaban	20	10	0
Skor	$20 \times 3 = 60$	$10 \times 2 = 20$	$0 \times 1 = 0$
Jumlah Skor	80		
persentase	$80/90 \times 100\% = 88,9\%$		

Garis Kontinum



Gambar 15. Garis kontinum kuisioner no. 5

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap menarik atau tidaknya tampilan aplikasi ini adalah 80 dari yang diharapkan 90, atau dapat dikategorikan setuju.

8. Kesimpulan

Metode forward chaining telah diterapkan pada pembuatan game finding selais. Konsep metode *forward chaining* pada game ini digunakan untuk proses *review* dan perpindahan level. *Rules* atau aturan yang digunakan pada metode *forward chaining* ini sebanyak 4 *rules*. Pengujian aplikasi ini menggunakan dua teknik yaitu: pengujian Alpha (pengujian dilakukan oleh pembuat aplikasi) melalui pengecekan

apakah aplikasi yang dibangun sudah sesuai dengan perancangan atau belum serta pengujian Beta (aplikasi diujikan kepada pengguna) dengan memberikan kuisioner yang terdiri dari lima pertanyaan terkait aplikasi *game finding selais*. Berdasarkan hasil pengujian terhadap aplikasi *game finding selais*, dapat ditarik kesimpulan aplikasi *game* dapat berfungsi sesuai dengan yang diharapkan serta aplikasi *game* yang dibangun telah cukup memenuhi tujuan awal pembangunan sebuah game.

9. Referensi

- [1] Putra, Y.S, M. Aziz Muslim, Agus Naba. "Game Chicken Roll dengan Menggunakan Metode Forward Chaining". Jurnal EECIS Vol. 7 No. 1, 2013.
- [2] Zainuddin. "Merancang dan Membuat Program Permainan Quad dengan Metode Forward Chaining". Skripsi STIKOM Surabaya, 2011.
- [3] Estrada, Andhika Hendra. "Metode Implementasi Pohon N-Ary dalam Artificial Intelligence Game Studi Kasus: Minimax pada Tic Tac Toe". Makalah pada Program Studi Teknik Informatika ITB, Bandung.
- [4] Andiansyah, Verdian. "Analisi Perancangan Game Desain Dokumen dan Daset Grafis dalam Pembuatan Game Otak Atik" Skripsi STMIK AMIKOM Yogyakarta, 2014.
- [5] Kusumadewi, Sri. "Artificial Intelligence (Teknik dan Aplikasinya)", Graha Ilmu, 2003.
- [6] Adam, Ernest. "Fundamentals of Game Design", Second Edition. Barkeley, CA: New Riders, 2010.
- [7] Nugroho, Eko. "Game Designer, Mitos atau Sungguhan?". link <http://tekno.kompas.com/read/2013/02/25/09282582/game.designer.mitos.atau.sungguhan>, diakses tanggal: 10 Agustus 2014.